

Videóvágás – minimalista módra

Ha az ember videót rögzít – legyen az TV felvétel vagy a születésnapis zsúr – óhatatlan, hogy felesleges részek is belekerülnek, amiket nem kívánunk megőrizni az utókor számára. Az ilyen részeket illik kivágni. Persze az is elképzelhető, hogy az anyag színvilága nem tetszik, de Linux alatt ezen is könnyen segíthetünk.

■ Egy igazi *Linux* guru képes akár parancssorból is vágni filmet *mencoder*-rel. Persze ilyenkor nem kell meglepődni, ha a *Windows*-on vagy *MacOS X*-en nevelkedett családtagok vagy kollégák a mentőket akarják hívni. Ennek egy fokkal kulturáltabb megoldás, amit jómagam is alkalmaztam egész sokáig: futtassuk az *ingyenesen használható Virtualdub*-ot vagy *Nandub*-ot *Wine* segítségével. Ez is csak ideig-óráig megoldás, hiszen sok *kodek szükséges* ezekhez a programokhoz, hogy akármilyen videófájl be tudjanak tölteni, vagy el tudjanak menteni. Már-már reménytelennek tűnt a dolog, de feltűnt a színen az *avidemux2*...

Gépigény

Az *avidemux2* gépigénye nem mondható túlzottnak, hiszen egy mai alsó-középkategóriás gépen is már elfut. Alsó-középkategóriás gép alatt jelen esetben egy minimum 1 GHz-es 256 megabájt memóriával és *AGP*-s videokártyával szereltet érték. Nyilván processzorból érdemes minél komolyabbat használni, mert különben gyakori lesz a „homokóra”. No nem a szerkesztés közben, hanem amikor végrehajtja a vágásokat a program és elmenti a végleges videót. Nem mondom processzor márkát, hogy melyik jobb a videóvágáshoz. Mindenki nyilván olyat fog venni, amiben hisz és ami még belefér a költségvetésbe, de egy jó tanács: komolyabb processzorra érdemes belevágni, hiszen egy normál *Intel* vagy *AMD*

processzor akár kétszer-háromszor is gyorsabb lehet ezek butított változatához képest (például *Celeron*, *Sempron*). A különbséget a processzor cache mérete adja, amely kb. 4-8-szoros (128/256 kbyte vs. 512/1024 kbyte) szokott lenni.

Illetve a másik dolog, amit jó ha tud a kedves Olvasó: hiába vett két- (vagy több) magos processzort, az *avidemux2* csak egy magot használ (jelenleg). Viszont nagy előny, hogy ilyenkor a többi magot másra használhatjuk.

Telepítés

Az *avidemux2*-t nem szokták mellékelni a *Linux* telepítő *CD*-khez, hiszen folyamatosan változik, és itt az is igaz, hogy nem mindegy, mennyire van gépünkre optimalizálva a program, hiszen a régebbi architektúrák nem támogattak bizonyos parancsokat, amelyek már az újabb processzorokban jelen vannak. Éppen ezért ha modern processzorra dolgozunk, nem biztos, hogy optimális futást kapnánk egy *Pentium 2*-re szánt bináris programmal. (A mai disztribúciók többsége *Pentium 2*-re van optimalizálva.) Jelen cikkben a 2.1.2-es (2006. március 1.) verziót fogom bemutatni, hiszen ez tekinthető az utolsó hivatalos kiadásnak, a többi egyelőre *preview* állapotban van.

Noha a forráskód kis méretű – a projekt honlapjáról csupán 4-5 megabájtot kell letöltenünk (kicsomagolva se sokkal több 20 megabájt nál), a fordításhoz már legalább 250 megabájt

üres területre lesz szükségünk.

A telepítés a szokásos *configure*, *make*, *make install* trióval elvégezhető, azonban pár csomagot telepítenünk kell előtte (Debian alatt: *pkg-config*, *libgtk2.0-dev*, *libsmjs-dev*, *libxml2-dev*), amelytől függ a program. Persze ezenkívül fel lehet telepíteni további csomagokat is, de azok nem feltétlenül szükségesek a program fordításához. Az elérhető funkciókat a *configure* parancs a végén listázza ki.

Amennyiben a *jsapi.h* fájl a *libsmjs-dev* csomag telepítése után sem találja a *configure* parancs, úgy használjuk a *-with-jsapi-include=<DIR>*, ahol a *<DIR>* az a könyvtár, ahol a *jsapi.h* található.

Amit érdemes feltelepíteni pluszként az adott projektek honlapjairól: *XVID*, *OSS* és/vagy *ALSA* támogatás, ha menet közben szeretnénk hangot hallani, *SDL* támogatás, *xvideo* támogatás a gyors kezelőfelületért, *lame* és/vagy *OGG* a hangtömörítésért. Nyilván ezekből nem elég a *lib* csomag, a fordításhoz szükséges az adott csomag fejlesztői változatai is (például *Debian*, *Ubuntu*, *UHU*, stb. esetén). Ha forrásból telepítettük ezeket, akkor legtöbbször az *avidemux2 configure* parancsának meg kell adnunk, hogy az adott függvénykönyvtár hol található.

Alapok

A program telepítés után az *avidemux2* paranccsal indítható.

A program nyelve a *LANGUAGE* környezeti változó segítségével állítható be.

```

Checking cpu/system type
CPU type : x86_64
ADM64
checking for AltiVec ... no
cpu done
GTK+ version      : 2,6,4
Dixv5,0.5 ?      : yes
*** DISABLED ***
Xvid 0.9 Codec    : no
Xvid 1.0 Codec    : yes
x264 encoder c   : no
Mad decoder      : yes
OGS              : yes
ALSA             : yes
SIL              : yes
Xvideo           : yes
Lane             : yes
Ogg Vorbis       : yes
Faad2(decoder)   : no
Faac (encoder)   : no
A52/AC3         : yes
FFMPEG          : yes
ESD              : yes
aRTs             : yes
FreeType         : yes
XML2             : yes
locale           :
Little endian cpu : yes
AMD64            : yes
AltiVec          : no
PPC              : no
X86              : 1
MacOsX(Darwin)  : no
Cygwin           : no
Gcc supports vectors:
Debug            : 0
configure: creating ./config.status
config.status: creating Makefile
config.status: creating admLavcodec/Makefile
config.status: creating admLavcodec/i386/Makefile

```

1. ábra Így néz ki egy avidemux2 konfigurálás, ha az ajánlott kiegészítők is felkerültek

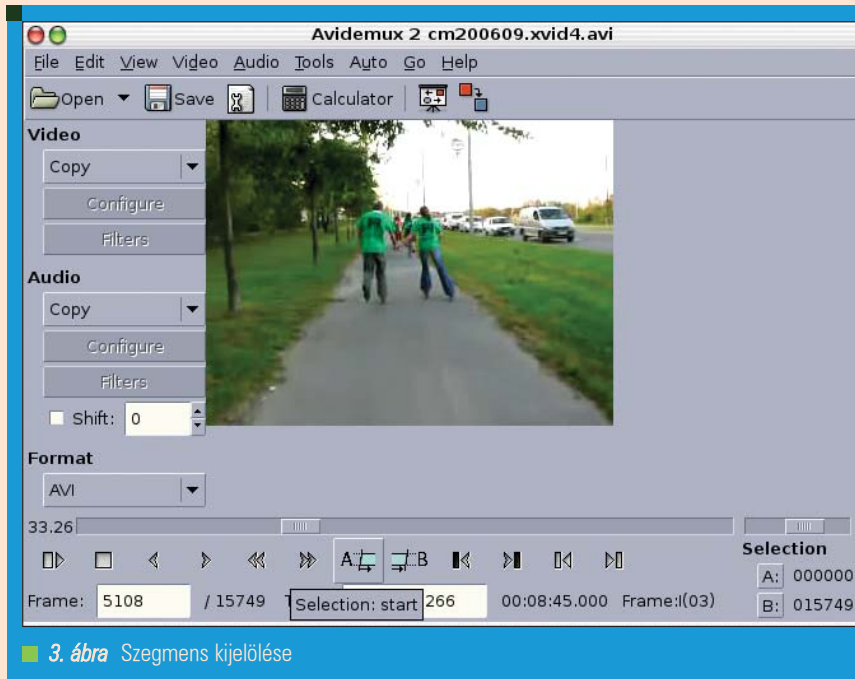


2. ábra Információk a megnyitott videóról

A 2.1.2 verzió esetén sajnos még nem tud teljesen magyarul a program. A program – mint már említettem – nem professzionális vágóprogram. Inkább kisebb vágásokra, filmek újratömörítésére ajánlanám. A filmekkel kapcsolatos beállítások és műveletek három csoportra oszthatóak: vágás, szűrők (*filters*) beállítása, tömörítés.

Vágás

Jelenesetben a vágás alatt például azt értem, hogy ha videóról digitalizálunk, akkor képkockára pontosan nem tudjuk elkapni a film elejét és a végét. Az *avidemux2* ezt nagyon egyszerűen meg tudja oldani: kijelöljük a szegmens elejét (A) és a végét (B), majd a szokásos vágólap műveleteket alkal-



3. ábra Szegmens kijelölése

mazhatjuk, jelen esetben a kivágást/törlést. Egérrel is könnyen kezelhető a program, de tapasztalatom szerint a billentyűzettel sokkal pontosabb és gyorsabb a pozicionálás. A billentyűzet jobb oldalán található számbillentyűzet használható (ne felejtjük el benyomni a *NumLock*-ot). A 2-es és a 8-as gombokkal a kulcsképkockák között tekerhetünk, míg a 4-es és a 6-os gombokkal egy-egy képkockát. A szögletes zárójelekkel – „[„ és „]” – rögzíthetjük az aktuális szegmens elejét és végét (3. ábra).

A legtöbb filmtömörítési megoldás kulcsképkockákkal (*keyframe*) dolgozik. Röviden az elv: teljes képkockát csak bizonyos képkockák után mentek le – például megadott időközönként, vagy ha túl sokat változott a kép – a közben lévő képkockákat mindig az előtte lévő kulcsképkockából számolják ki a programok. Éppen ezért ne lepődjünk meg, ha visszatekerésekor kicsit lassabb a program, mint előretekeréskor.

Szűrők beállítása

A szűrők beállítása előtt a videókodeket állítsuk át valamire, teljesen mindegy. Ha nem így teszünk, akkor nem férünk hozzá a szűrők dialógus ablakához. Az szűrők között többféle található. A legtöbbel csupán minimális módosításokat tudunk megejtetni, például alakíthatunk a videó színvilágán, élesíthetjük a képet, szélesvásznúra alakíthatjuk, stb.

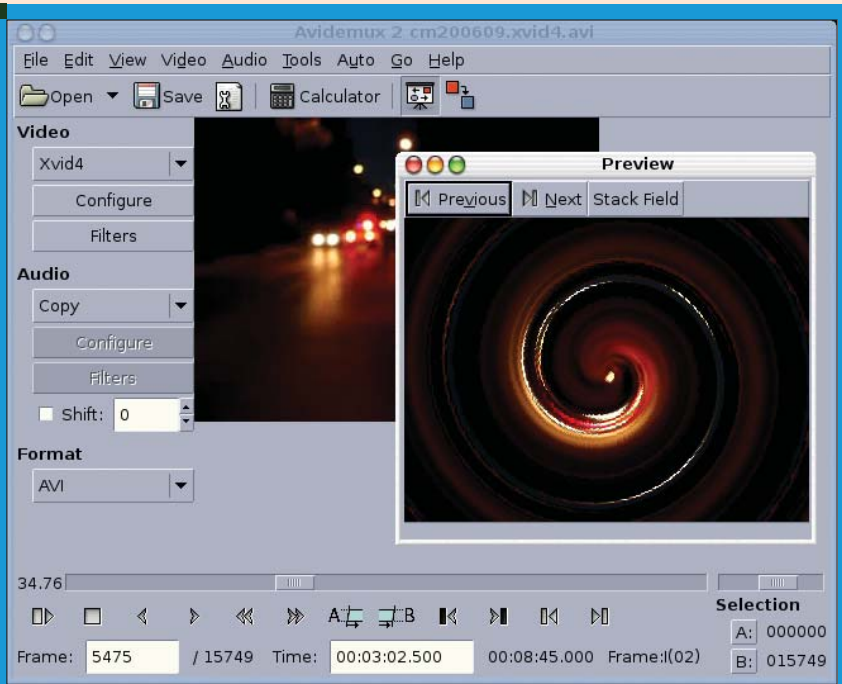
Van azonban olyan szűrő is, amellyel létfontosságú korrekciókat hajthatunk végre. Ilyen például a *deinterlacer*. Ha például teljes felbontásban (704x576 képpont PAL esetén) veszünk fel mondjuk *TV*-ből, akkor nagyobb mértékű vízszintes mozgásoknál a kép páros és páratlan sorai szétcsúsznak, fésűs lesz.

A *deinterlacer* megfelelő matematikai műveletekkel ezt tudja korrigálni. Amennyiben azonban kisebb felbontásban veszünk fel (352x288 képpont *PAL* esetén), úgy felesleges *deinterlacert* bekapcsolni.

A szűrők között találunk például olyat is, amellyel *srt* formátumú feliratot „égethetünk” a videóra.

Érdekes, ámde többé-kevésbé hasznontalan szűrő, amely elméletileg a tv csatornák logóit hivatott eltüntetni. A gyakorlatban ez csak halványítást jelent, hiszen a logó alatti részt nem találja, nem találhatja ki az algoritmus. A filterek sorrendje, lévén, hogy az ember ritkán választ több hasonlót egy munkához, lényegtelen, azonban vannak esetek, amikor nem mindegy, hogy a listában melyik hol van (7. ábra).

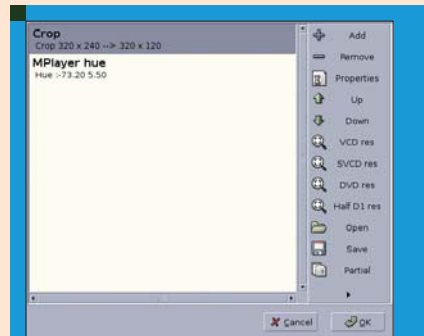
Ha elkészültünk a szűrők összeválogatásával és beállításával, nyomjuk meg az *F5*-öt (előnézeti ablak) és a csúszka alatt nyomjuk meg a lejátszást. Ekkor az előnézeti ablakban látható, hogy néznek ki az szűrők együtt.



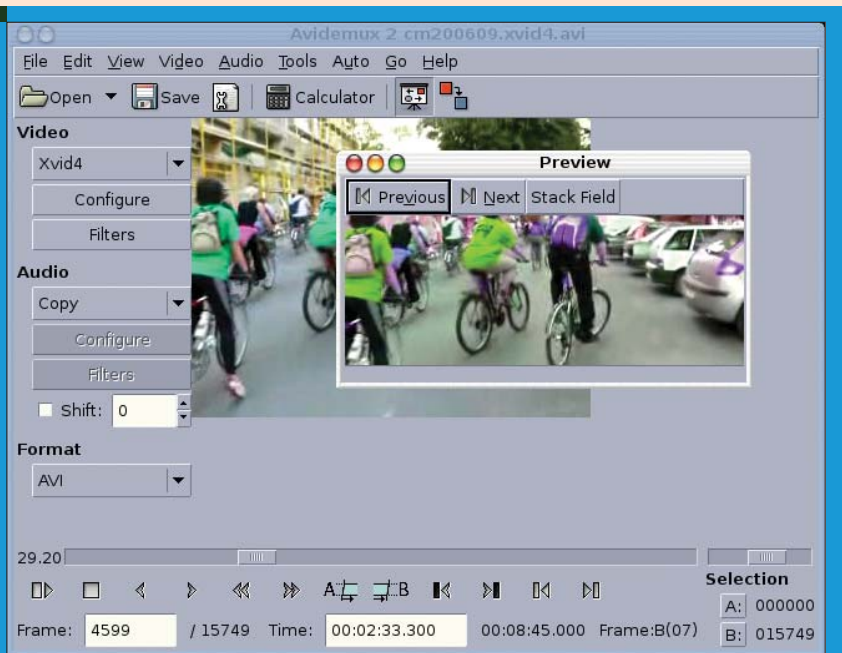
4. ábra Spirál szűrő és előnézete



6. ábra Elérhető szűrők listája



7. ábra Használatban lévő szűrők listája



5. ábra Színkorrekció



8. ábra Hang átalakítás

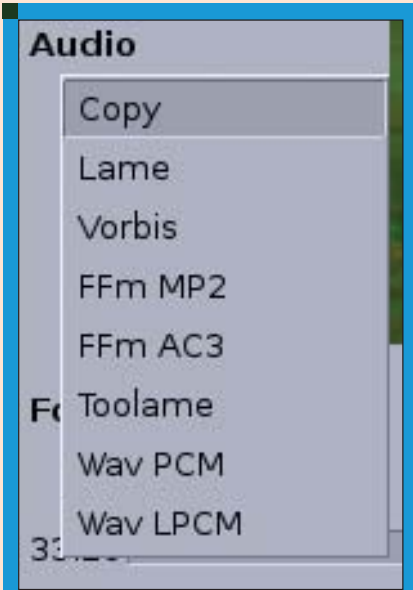
Tömörítés

Ha a vágás és a szűrők kiválasztásával megvagyunk, akkor itt az ideje elmenteni a végeredményt. Azonban egy jó tanács: mentjük el a projekt fájlt az elején. Ez az állomány írja le a vágásainkat és a szűrőket, illetve azok beállításait. A mentést azért javasolom, mert egy esetleges áramszünet esetén nem kell előlről kezdeni, már lesz egy projekt fájl, amit csak be kell tölteni.

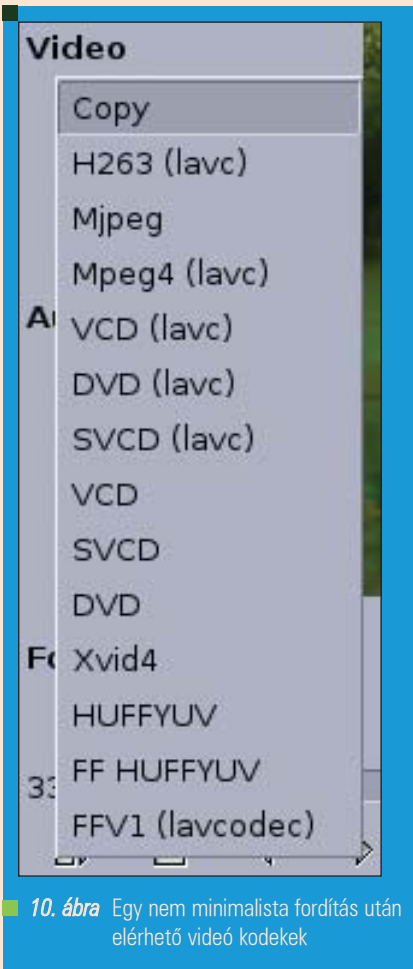
A tömörítés kodekenként különböző opciókat ad, de csak pár fontosabbat emelnék ki, hiszen a többi többnyire optimális, de akinek nem felel meg, az próbálgathatja.

Először jelöljük ki a használni kívánt audió és videó kodeket. Ha nincs a kedvenc kodekünk a listában, akkor töltsük le az adott projekt honlapjáról és fordítsuk újra az *avidemux2*-t is.

A videót (12., 13. ábra) jellemzően kétféleképp tömöríthetjük: fix vagy változó sávszélességgel. A fix sávszélesség előnye: hamarabb végez vele a kedves Olvasó, de nem ad optimális képminőséget. A változó sávszélesség optimális eredményt ad ugyan, de a program kétszer megy végig a videón. Az első alkalommal meghatározza, hova kell több illetve kevesebb bit. (Például egy autós

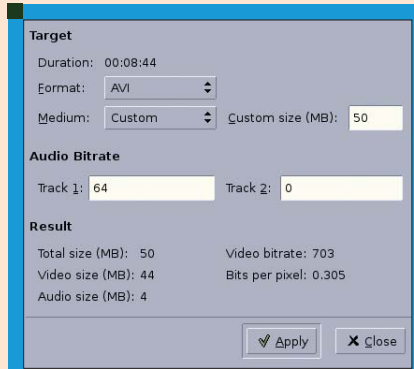


9. ábra Egy nem minimalista fordítás után elérhető hang kodekek

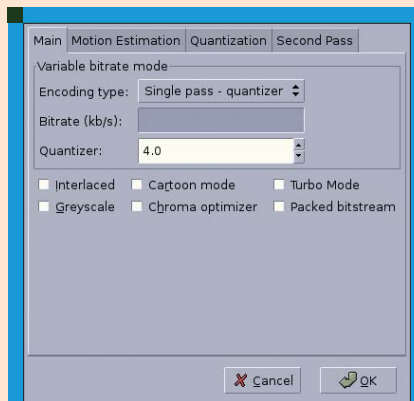


10. ábra Egy nem minimalista fordítás után elérhető videó kodekek

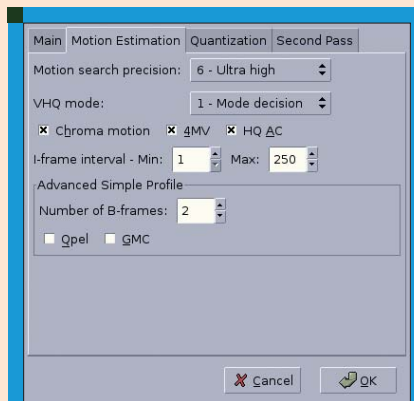
üldözésnél nagyobb sávszélesség szükséges, különben kockás lesz a kép, míg egy csendélethez alig kell valami, ott szinte nincs változás a képkockák között.) Második alka-



11. ábra Beépített számológép, amellyel egyszerűen kiszámolható méret alapján a videó sávszélessége



12. ábra Xvid tömörítés beállításai



13. ábra Xvid tömörítés haladó beállításai

lommaal pedig végrehajtja a tömörítést az első alkalommal letárolt statisztikai fájl segítségével. Az egy menetes tömörítésnél jellemzően sávszélességet vár a program, míg a két menetesnél jellemzően várt adat a fájl méretet. A hangtömörítés (14. ábra) kicsit könnyebb dolog, hiszen ott 128 kbites MP3 már többé-kevésbé kiváló hangminőséget eredményez.



14. ábra Hangtömörítés beállításai

Végül a *Format*-nál kiválaszthatjuk, hogy milyen formátumban szeretnénk elmenteni a végeredményt. Ha *AVI*-n hagyjuk, akkor az esetek többségében megfelelő, kivéve, ha *MPEG*-et szeretnénk készíteni, mondjuk *DVD*-re. Hasznos választás még az *AVI dual channel*. Ez lehetővé teszi, hogy egy videóhoz két hangsávot tároljunk el. Ezeket az *Audió* menüben állíthatjuk be. Ha mindennel megvagyunk, indítsuk el a mentést. Ekkor a program végrehajtja a vágásokat, szűrőket, tömörítéseket. A művelet a videó hosszától, az alkalmazott szűrőktől, a gép sebességétől és a tömörítési beállításoktól függően akár több órát is igénybe vehet.



Medve Zoltán
(e-medve@e-medve.hu)

2001-ben kezdett „Linuxolni”, de már korábban is ismerkedett a szabad szoftverek világával. Ha éppen nem a gép előtt ül, akkor fotóztat, olvasgat vagy bicajozik.

KAPCSOLÓDÓ CÍMEK

Az Avidemux2 honlapja:
➔ <http://www.avidemux.org/>

Az Ogg honlapja:
➔ <http://www.xiph.org/>

Az Xvid honlapja:
➔ <http://www.xvid.org/>

A Lame honlapja:
➔ <http://lame.sourceforge.net/>